

Docket AUS920000510US1

Appl. No.: 09/736,349
Filing Date: December 14, 2000**REMARKS****OBJECTIONS DUE TO MINOR INFORMALITIES**

Amendments are herein submitted, as set out above, to correct typographical errors in claims 17 and 23 regarding "for" and "to," respectively.

REJECTIONS UNDER 35 U.S.C. 112, SECOND PARAGRAPH

An amendment is herein submitted, as set out above, to correct claim 2 so that it depends on claim 1. (An amendment is also herein submitted to correct claim 13 so that it depends on claim 9, as was originally intended.) Amendments are herein submitted, as set out above, to change claims 5, 7, 13, 15, 21, and 23 to state "at least one of the web pages" instead of "at least some of the web pages." Applicant contends that these amendments overcome the rejection under 35 U.S.C. 112, second paragraph.

AMENDMENTS DUE TO AN ISSUE NOTICED BY APPLICANT

It is implied that a computer program product, as set out in claims 9 and 15 of the present application, resides on a computer usable medium having computer readable program code. This is stated in the specification. Present application, page 17, lines 5-12. Nevertheless, amendments are herein submitted to claims 9 and 15, as set out above, to explicitly state that each claimed computer program product resides on a computer usable medium having computer readable program code. This is in order to ensure the claims fall within the statutory subject matter requirements of 35 U.S.C. 101. No new matter is added in the amendments to claims 1, 9 and 17, since the specification as originally submitted provides support, as indicated above.

REJECTIONS UNDER 35 U.S.C. 102(e)

Claims 1, 4, 9, 10, 12, 17, 18 and 20 are rejected under 35 U.S.C. 102(e) as being anticipated by United States Patent No. 6,301,614 ("Najork"). Presumably, the rejection also applies to claim 2, as explained herein below.

Claims 1, 9 and 17

Applicant respectfully contends that claims 1, 9 and 17 are patentably distinct, for reasons explained herein below. Moreover, Applicant amends claims 1, 9 and 17, as set out above, to even more distinctly distinguish the present invention.

Docket AUS920000510US1

Appl. No.: 09/736,349
Filing Date: December 14, 2000

Najork describes the problem it addresses as follows:

A web crawler is a program that automatically finds and downloads documents from host computers in an Intranet or the world wide web . . . Before the web crawler downloads the documents associated with the newly discovered URL's, the web crawler needs to find out whether these documents have already been downloaded . . . Thus, web crawlers need efficient data structures to keep track of downloaded documents and any discovered addresses of documents to be downloaded. Such data structures are needed to facilitate fast data checking and to avoid downloading a document multiple times.

Najork, col. 1, lines 34-61. Specifically, the teachings of Najork concern "the data structures and methods used to keep track of the URL's of documents that have already been downloaded or that have already been scheduled for downloading." Najork, col. 4, lines 54-57.

The web crawler taught by Najork includes "threads 130 for downloading web pages from the servers 112, and processing the downloaded web pages; a main web crawler procedure 140 executed by each of the threads 130; and a URL processing procedure 142 executed by each of the threads 130 to process the URL's identified in a downloaded web page." Najork, col. 3, lines 31-58. Each thread executes a main web crawler procedure 140 shown in FIG. 3. Najork, col. 4, lines 58-59. The web crawler thread determines the URL of the next document to be downloaded (step 160) and then downloads the document corresponding to the URL, and processes the document (162). Najork, col. 4, lines 59-64. According to that processing, the main procedure identifies URL's in the downloaded document that are candidates for downloading and processing (step 162). Najork, col. 4, line 66 - col. 5, line 3.

Najork specifically points out that "these URL's are typically found in hypertext links in the document being processed." Najork, col. 5, lines 3-4. But, as particularly pointed out in the present application, sometimes URL's are *not* found in hypertext links, which presents a problem. That is, one reason a conventional crawler and the crawler taught by Najork are not suitable for the "staticizing" problem addressed in the present invention is that "references from one web page to another may not be straightforward. That is, a reference may not be simply set out on the page as a hyperlink address [i.e., a URL], but instead may be a script, form, selection menu, or button for example. Thus a need exists for improvements in crawler programs, to overcome their limitations so that they may be used for the staticizing problem as well as other applications."

Docket AUS920000510US1

Appl. No.: 09/736,349
Filing Date: December 14, 2000

Present application, page 2, line 20 - page 3, line 6. Najork offers no teaching that addresses this problem, or even that suggests it exists.

The present application further elaborates on the problem, explaining how a reference that is not "simply set out on the page as a hyperlink address, but instead . . . specified by a script, for example, so that the address is produced only when a client browser executes the reference."

Present application, page 5, lines 10 - 15; see also, page 12, line 21- page 13, line 1 (describing how the reference may be specified by a script, a selection menu, form, button or other element).

The present application goes on to explain how this problem may be addressed, as follows:

To generate references of this sort in connection with generating the requests to the server, another aspect of the invention arises. According to an embodiment, the crawler parses each received web page and sends references to an applet developed for an embodiment of the present invention that runs in the browser. (This applet may be referred to herein as a "JavaScript execution engine" or simply "JEE.") The browser determines the address for a next page responsive to such a reference, so that the browser may receive the next page and any cookie for the next page from the server, and the JEE returns the address and any cookie to the crawler program.

Present application, page 5, lines 15-22; page 15, lines 2-8. Claims 1, 9 and 17 particularly point this out, as well.

Claim 1, as amended, states that ". . . a page has a reference, wherein the reference is specified by a script for producing an address for a next page." (Due to the introduction of "executing" in the last step of the claim, the language about "executing" is deleted from this first step.) The claim goes on to say that such a reference is parsed from one of the web pages by the crawler program and sent to an applet running in a browser. Further, the claim is herein amended to clearly tie the pieces together by stating that the address for the next page is determined by the browser "executing" the reference and sending the address to the crawler. Claims 9 and 17, as amended, have similar language.

No new matter is added in the amendments to claims 1, 9 and 17, since the specification as originally submitted provides support. Present application, page 5, lines 10-22 (regarding the reference having a script that is executed by a browser for the crawler).

The present application specifically states that according to the present invention "at least one page of the web site has a reference for *executing* by a browser to produce an address for a

Docket AUS920000510US1

Appl. No.: 09/736,349
Filing Date: December 14, 2000

next page.” Present application, claims 1, 9 and 17 (emphasis added). Likewise, Applicant intended for this language to clearly point out something different than merely finding an address that is explicitly set out in a hypertext link, which is what is taught by Najork. Consider the following examples which further illustrate the difference.

An example of a hypertext link that explicitly has the text of an address set out therein, as alluded to by Najork, is as follows: ``. In contrast, the following hypertext link provides an example of a reference that is not so straightforward and that is “specified by a script to produce an address” so that the address for the next page is determined “by the browser executing to the reference,” as stated in amended claim 1 of the present application: ``. See Web page, http://freshair.npr.org/day_fa.jhtml?display=day&todayDate=09/21/2004. Executing this reference produces a URL such as the following:

<http://www.npr.org/dmg/dmg.html?prgCode=FA&showDate=22-Sep-2004&segNum=1&NPRMediaPref=RM>.

The present application says “a reference . . . may be specified by a script” because an href tag, for example, typically has a *call* to a script and not the script *itself*. The browser locates the source code for the function that is called and then executes the specified function.

Note also, a “reference” is not limited to the context of an href tag. Consider the following example snippet of HTML code:

```
<form>
  <input type="button" value="GO" onclick="DoSearch()"/>
</form>
```

This snippet creates a button that says “Go.” When the user presses the button the browser needs to execute the function `DoSearch()` in the context of the button before it can determine what URL to load. In this example also the URL to be loaded is “specified by a script,” the `DoSearch()` script, which is not itself included in the form that produces the button.

Note also the amended claim, like the passage of the specification set out above, states that the browser executes the *reference* instead of saying merely that the browser executes the script. In the snippet example above, the crawler needs to know what URL to load when the button is pushed. The crawler achieves this by telling the browser (via the applet) to push the button. It cannot tell the browser to just execute the JavaScript function “`DoSearch()`” because

Docket AUS920000510US1

Appl. No.: 09/736,349
Filing Date: December 14, 2000

the browser would then not have the context in which to execute the function. See present application, page 5, lines 10 - 15 (explaining that the address "is very dependent on the context in which it is produced, that is, the history that led up to it, including the state of the server and the client browser."); see also, page 15, lines 2-8 (explaining that the crawler passes information 230 to a JavaScript execution engine 210 for generating queries to the web server 100 and that the information includes the JavaScript command that invokes script 303 when button 304 is clicked, a context object, the browser window object, and the document object associated with page 140.X in its context as it exists, loaded in browser 205).

Applicant recognizes that *executing* a reference to produce an address, as claimed, might be confused with *parsing* the reference to find an address that is explicitly set out therein. The explanation above clarifies these significant differences. Also, to make the distinction particularly clear in the claims, Applicant herein submits the amendments described above to claims 1, 9 and 17.

It should be clear from the discussion above that the amended claims are patentably distinct from Najork, col. 4, line 59 - col. 5, line 4, which the Office action relies upon for the rejection. For these reasons Applicant contends that claims 1, 9 and 17 are allowable.

Claims 2, 10 and 18

The Office action relies upon Najork FIG. 1, domain name system 114 for the rejection of claims 10 and 18. Applicant assumes that claim 2 is likewise rejected on this basis, since claim 2, a method form of the invention, sets out a feature similar to what is set out in claims 10 and 18, which are different forms of the invention. Applicant herein amends claims 2, 10 and 18, as set out above, to clearly distinguish the present invention over a conventional domain name server arrangement.

No new matter is added in the amendments to claims 2, 10 and 18, since the specification as originally submitted provides support for the amendments. Present application, page 6, lines 1-14.

As explained in the present application, due to the JavaScript execution engine ("JEE") used in an embodiment of the present invention to deal with JavaScript in hypertext references, a further difficulty arises because an applet running on a client browser can, for security reasons, only interact with objects in a web page if the web page and the applet are loaded onto the client

Docket AUS920000510US1

Appl. No.: 09/736,349
Filing Date: December 14, 2000

from the same server. Present application, page 6, lines 1-4; page 15, line 9 - page 16, line 2. To overcome these limitations, the client browser is configured to use a certain proxy gateway and to a certain file (referred to herein as the "resolver file") containing cross-referencing for hostname-to-IP-address-resolution. Present application, page 6, lines 6-9; page 15, lines 15-17. The resolver file indicates the IP address of the proxy gateway as the IP address for the web site server, even though the proxy gateway is not really the source of the web site server (hence the proxy gateway is referred to as a "spoof proxy"), so that the JEE running on the client *appears* to the client's browser to be from the same server as the web pages. Present application, page 6, lines 10-14; page 15, lines 17-23. This permits the JEE to communicate with the crawler and the browser unhindered by conventional limitations. Page 15, line 22 - page 16, line 2.

Consistent with the above, the present application specifically states that according to the present invention, "the browser [is] configured to use a certain proxy, and refer to a resolver file for hostname-to-IP-address-resolution, and wherein the web site server has an IP address, the proxy for the browser has a certain IP address, and the resolver file indicates the certain IP address as the IP address for the web site server." Present application, claims 2, 10 and 18 (emphasis added).

The Office Action compares the arrangement described in the claims with a conventional DNS server arrangement, as described in the cited figure of Najork. However, a proxy gateway is not really the source of a web site server, i.e., the proxy gateway and the web site server have different IP addresses. Therefore, a DNS server would not have an arrangement in which "the resolver file [indicates] the certain IP address as the IP address for the web site server," as claimed. Nevertheless, Applicant recognizes that the claims do not explicitly rule out the possibility that the proxy and the web site server have the same address, in which case a DNS server would indicate the IP address of the proxy as the IP address for the web site server. Applicant recognizes that the Office action may rely on this point for a contrived comparison, as contrary to logic as this posited arrangement would ordinarily be. Therefore, to make the distinction even more particularly clear, Applicant herein submits amendments to claims 2, 10 and 18 stating that "the certain IP address of the proxy [is] different than the IP address of the web site server." Thus, the amended claims indicate that even though the proxy and the web site

Docket AUS920000510US1

Appl. No.: 09/736,349
Filing Date: December 14, 2000

server have different addresses, they are resolved as the same address. Clearly this is not taught or suggested by Najork or any of the other cited references.

For the above reasons Applicant contends that claims 2, 10 and 18 are allowable. Further, Applicant contends that the claims are patentably distinct because they respectively depend upon allowable claims.

Claims 4, 12 and 20

Claims 4, 12 and 20 state that the crawler is programmable to perform particular action sequences for generating the queries to the web server. For the rejection of these claims, the Office action relies upon Najork, col. 4, lines 59-62, which is discussed herein above in connection with claim 1. That is, the Office action equates the "action sequences" of the claims in the present application to the teaching by Najork about a web crawler thread determining the address of a next document to be downloaded by parsing a URL that is set out in a hypertext link, downloading the document corresponding to the URL, and processing the document, as described by Najork, col. 4, lines 59-64.

In the present application, the claimed action sequences are different than the mere selection of hypertext links and parsing of URL's described in the above passage of Najork. The application explains action sequences as follows:

In one aspect of an embodiment of the crawler of the present invention, the crawler is programmable to perform particular actions sequences for generating queries to the web server. To clarify, consider an example of actions performed by a user to obtain a particular end data set using a conventional web page form.

Referring now to FIG. 3, Web page 140.X is shown in further detail. The page 140.X has two lists 301 and 302, for selecting parameters for generating a query. First list 301 is for selecting a state. Second list 302 is for a profile. In the example, in a conventional web page access, where a user is controlling browser 205, if the user wants to obtain income information for the state of Texas from a Community Facts page in the American Fact Finder web site (factfinder.census.gov), the user performs the following action sequence 305:

1. Select Texas as the State from list 301.
2. Select Income as the Profile from list 302.
3. Click on the create button 304, which causes a script 303 to query server 100 with a request that includes the parameters selected from the lists 301 and 302.

The point to note is that obtaining the desired data requires these actions to be performed, in the proper sequence. Conventional crawlers have not been programmed to do this. Accordingly, in one aspect of an embodiment of the crawler 171 of the present invention, the crawler is programmed as shown at 310

Docket AUS920000510US1

Appl. No.: 09/736,349
Filing Date: December 14, 2000

to perform particular actions sequences for generating information 230 to pass to the JEE 210 for generating queries to the web server 100. In the example, the information 230 includes the JavaScript command that invokes script 303 when button 304 is clicked. Also including in the information 230 is a context object, the browser window object, and the document object associated with page 140.X in its context as it exists, loaded in browser 205.

Present application, page 14, line 10 - page 15, line 8.

To make the distinction regarding the present invention more clear, claims 4, 12 and 20 are herein amended, as set out above, to state that "the crawler is programmable to perform particular action sequences for selecting non-hypertext-link parameters from the at least one web page in a particular sequence, so that the queries to the web server include the selected parameters and a context arising from the particular sequence." The amendment makes it clear that the claims refer to an action sequence such as that described in the passage set out immediately above, in which non-hypertext-link parameters, such as those in pull-down lists and the like, are selected in a particular sequence. Clearly this is not the sort of sequence taught or suggested by Najork or any of the other cited references.

No new matter is added for these amendments, since the specification as originally submitted provides support, as described in the passage set out immediately above. See also, present application, page 12, line 21 - page 13, line 8 (discussing the importance of context in generating queries).

For the above reasons Applicant contends that claims 4, 12 and 20 are allowable. Further, Applicant contends that the claims are patentably distinct because they respectively depend upon allowable claims.

Docket AUS920000510US1

Appl. No.: 09/736,349
Filing Date: December 14, 2000REJECTIONS UNDER 35 U.S.C. 103(a)Claims 5-8, 13-16, and 21-24

Claims 5-8, 13-16, and 21-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Najork in view of Challenger. Applicant contends that dependent claims 5, 6, 13, 14, 21 and 22 are patentably distinct at least because they each depend on respectively allowable claims. MPEP 2143.03 ("If an independent claim is non obvious under 35 U.S.C. 103, then any claim depending therefrom is non obvious," citing *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)). Independent claim 7 is amended herein to incorporate the limitations of original dependent claim 8. Likewise, independent claim 15 is amended to incorporate the limitations of original dependent claim 16 and independent claim 23 is amended to incorporate the limitations of original dependent claim 24. Applicant contends that independent claims 7, 15 and 23, as amended, are patentably distinct for reasons explained herein below.

As explained herein above in connection with claims 4, 12 and 20, the term "action sequence" is used in the present application to describe selecting non-hypertext-link parameters, such as those in pull-down lists and the like, in a particular sequence. In the example discussed above in connection with claims 4, 12 and 20, a user viewing a web page performs an action sequence to get Texas State income information from a Community Facts page in the American Fact Finder web site (factfinder.census.gov). Specifically, a user may select the non-hypertext-link elements: "Texas" in list 301, "Income" in list 302, and a "create" button, in that particular order. This action sequence causes a script to query a server with a request that includes the parameters selected from the lists 301 and 302. As also explained above, it is an advantage of the present invention that a crawler is programmed to perform these action sequences, that information 230 is generated, and that the information 230 is passed to execution engine 210 for generating queries to the web server 100.

In claims 5-8, 13-16, and 21-24, the focus is on features of the *web page* itself, i.e., "operations," that are *subjected* to such an action sequence and that are replaced by references, such as conventional hypertext links. The present specification states:

It should be understood that web page 140.X includes the results of data dynamically generated in server 100 for an earlier query 250 that resulted in the page 140.X, but 140.X also has *operations* included in it *linking* it to page 140.Y, that call for the server 100 to generate more dynamic data to produce web page 140.Y. Therefore, web page 140.X needs to be converted by removing these

Docket AUS920000510US1

Appl. No.: 09/736,349
Filing Date: December 14, 2000

operations and replacing them with *references* to a "version of" the returned web page 140.Y. The reference must be to 140.Y1, a "version of" web page 140.Y, because 140.Y may itself have operations included linking it to a next page also calling for the server 100 to generate more dynamic data to produce the next page.

Present application, page 13, lines 16-23 (emphasis added). In this regard, amended claim 7 states that a web site server is queried by a crawler program responsive to references from one web page to another in the web site, wherein the queries are for causing the server to generate web pages and at least some of the web pages are dynamically generated. The server generated web pages are processed to generate corresponding processed versions of the web pages, so that the processed versions can be served in response to future queries, reducing dynamic generation of web pages by the server. At least a first such server generated web page has included in it *an operation* that would cause the server to dynamically generate a second web page if the first page were used to generate further requests to the server. The processing of the server generated web pages includes removing the operation from the first server generated web page and *replacing the operation with a reference to a version of another of the server generated web pages*. Claims 15 and 23, as amended, have similar language.

Applicant recognizes, however, that in spite of any special meaning attributed to the term in the present specification, the Examiner reads the term "operation" according to its plain-language meaning. Therefore, Applicant herein further amends claims 7, 15 and 23 to state that the operation includes "a number of non-hypertext-link elements on the first page selected in a particular sequence." No new matter is added for these amendments, since the specification as originally submitted provides support. Present application, page 14, line 10 - page 15, line 8 (describing the non-hypertext link elements: "Texas" in list 301, "Income" in list 302, and a "create" button, in that particular order).

Amended claims 7, 15 and 23 relate, for example, to an arrangement as described in the present application in which a server-generated top-level page, which a user can use to determine the incomes of people in 50 states, may have (1) a drop-down list of fifty states and (2) a GO button. When clicked, the GO button loads a page containing income information of a selected one of the States. The amended claims describe how this top-level server-generated page is processed to become a top-level page (the claimed "second" web page) that has a "reference" (which could be a URL link) pointing to a page that specifies the income of a selected one of the

Docket AUS920000510US1.

Appl. No.: 09/736,349

Filing Date: December 14, 2000

fifty States. Thus, when a user clicks on the URL in the new top-level page, no server activity is required as was the case for the dynamic web page (the claimed "first" web page) from which the second web page was created. It should be also understood that all fifty of the operations in the dynamic web page for the fifty states may be replaced in the claimed manner.

Applicant contends that the above described amendments patentably distinguish claims 7, 15 and 23. Applicant notes that Challenger does not teach or suggest about an operation that includes "a number of non-hypertext-link elements on the first page selected in a particular sequence," as claimed. Further, Challenger does not teach or suggest *removing* such an operation from a web page and *replacing* the operation with a reference to a version of another of the server generated web pages. Challenger describes FIG. 1C, which is relied upon for the rejection, as follows. Dynamically created web pages are cached, and application 97 specifies records the cached pages depend upon. Challenger, col. 10, lines 5-18. The application notifies cache manager 1 if a record changes, and the cache manager responsively updates pages in the cache that depend on the changed record. Challenger, col. 10, lines 18-24.

Claims 3, 11 and 19

Claims 11 and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Najork in view of Yoshida. Applicant assumes it was also intended that claim 3 be rejected on the same basis, since claim 3 is similar to claims 11 and 19. Applicant contends that dependent claims 3, 11 and 19 are patentably distinct at least because they each depend on respectively allowable claims. MPEP 2143.03.

Docket AUS920000510US1

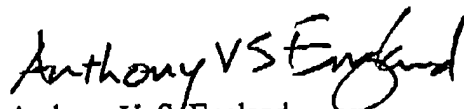
Appl. No.: 09/736,349
Filing Date: December 14, 2000**PRIOR ART OF RECORD**

Applicant has reviewed the prior art of record cited by but not relied upon by Examiner, and asserts that the invention is patentably distinct.

REQUESTED ACTION

Applicant contends that the invention as claimed in accordance with amendments submitted herein is patentably distinct, and hereby requests that Examiner grant allowance and prompt passage of the application to issuance.

Respectfully submitted,



Anthony V. S. England
Attorney for Applicants
Registration No. 35,129
512-477-7165
a@aengland.com